Université de Paris

Sémantique
Feuille n° 2 : λ-calcul

Exercice 1 : Manipulation

On considère ls termes suivants :

$$
\begin{array}{rcl}
I & = & \lambda x.x \\
K & = & \lambda x.\lambda y.x \\
S & = & \lambda x.\lambda y.\lambda z.x\, z\, (y\, z) \\
\Delta_f & = & \lambda x.f(xx) \\
Y & = & \lambda f.\Delta_f \Delta_f \\
\Delta & = & \lambda x.xx \\
\Omega & = & \Delta\Delta
\end{array}
$$

1. Quelles sont les variables libres et liées dans chacun de ces termes.
2. Parmi les termes suivants, les quels sont α-équivalents : $\lambda x.\lambda y.y$, $\lambda y.\lambda x.x$, $\lambda x.\lambda x.x$, $\lambda x.xy$ et $\lambda y.yy$
3. Calculer IK, KI, SKK, Ω, $K\ I\ \Omega$, $K\ u\ v$ pour u et v des termes quelconques.

Exercice 2 : Programmation

Utiliser votre langage de programmation préféré pour encoder les termes du lambda-calcul.

1. Définir un type ou une structure de données pour représenter les termes du lambda-calcul.
2. Définir une fonction d'affichage qui transforme votre encodage des termes du lambda-calcul en quelque chose de lisible. Vous utiliserez cette fonction pour tester les fonctions suivantes.
3. Définir la fonction `free_var` qui associe à tout terme la liste de ses variables libres
4. Définir la fonction `rename_var` qui renomme les variables d'un lambda-terme pour éviter les captures de variables (toutes les variables liées ont un nom différent et une variable liée ne peut pas avoir le même nom qu'une variable libre).
5. Définir la fonction substitution `subst` telle que `subst s x t` représente $s\{x/t\}$.

Exercice 3 : Induction sur les λ-termes

Utiliser l'induction pour prouver les propositions suivantes :

1. Si M et N sont des termes et x une variable telle que $x \notin FV(M)$ (x n'a pas d'occurrences libres dans M) alors $M\{x/N\} = M$
2. Lemme des substitution : Pour tous termes M, N, L et variables x, y t.q. $x \neq y$, si $x \notin FV(L)$ alors
$$M\{x/N\}\{y/L\} \ = \ M\{y/L\}\{x/N\{y/L\}\}$$
3. $FV(M\{x/N\}) \subseteq FV(M) \setminus \{x\} \cup FV(N)$.
4. Si $M \to_\beta N$, alors $\mathrm{FV}(N) \subseteq \mathrm{FV}(M)$.

Exercice 4 : Couples

1. Si M et N sont des λ-termes donnés, définir des termes $\langle M,N\rangle$, π_1 et π_2 tels que $\pi_1\langle M,N\rangle \to_\beta^* M$ et $\pi_2\langle M,N\rangle \to_\beta^* N$.
2. Définir un codage des booléens et du if_then_else en λ-calcul.

Exercice 5 : Arithmétique dans le λ-calcul

Soit $\overline{n} = \lambda x \lambda y.\underbrace{(x(x\ldots(x}_{n \text{ fois}}y)\ldots))$ une représentation des entiers naturels en λ-calcul. Définir :

1. Un terme S qui réalise la fonction successeur : $S\overline{n} \to_\beta^* \overline{n+1}$.
2. Un terme T qui itère n fois une fonction f à partir de l'entier m : $(T\overline{n})\overline{m} \to_\beta^* \underbrace{(f(f\ldots(f}_{n \text{ fois}}\overline{m})\ldots)$.
3. Un terme A qui réalise la somme de deux entiers : $(A\overline{n})\overline{m} \to_\beta^* \overline{n+m}$.
4. Un terme M qui réalise la multiplication de deux entiers : $(M\overline{n})\overline{m} \to_\beta^* \overline{n \times m}$.
5. Un terme E qui réalise l'exponentiation : $(E\overline{n})\overline{m} \to_\beta^* \overline{m^n}$.

Exercice 6 : Itération

1. Définir un terme P qui réalise la fonction prédécesseur, c.à.d. tel que pour tout entier naturel n,

$$P\overline{n} \to_\beta^* \begin{cases} \overline{0} & si\ n = 0 \\ \\ \overline{n-1} & sinon \end{cases}$$

2. Définir un terme F qui réalise la fonction de Fibonacci, c.à.d. tel que pour tout entier naturel n, $F\overline{n} \to_\beta^* \overline{fib(n)}$, où

$$fib(n) = \begin{cases} 1 & si\ n \leqslant 1 \\ fib(n-1) + fib(n-2) & sinon \end{cases}$$